Algorithm 1: Dataset logic: resampling, feature extraction, label tokenization and masking

```
class StreamingASRDataset(ds, processor, max_label_length = 410)
    // Store handles
    self.ds ← ds;  self.processor ← processor;  self.max_label_length ← 410;

    function __len__()
        return |self.ds|;

    function __getitem__(idx)
        // 1) Fetch sample
        sample ← self.ds[idx];
        audio ← sample["audio"]["array"];
        sentence ← lower(sample["text"]);
        // 2) Resample to 16 kHz if needed
        sr ← sample["audio"]["sampling_rate"];  target_sr ← 16000;
        if sr ≠ target_sr then
            audio ← librosa.resample(audio, orig_sr=sr, target_sr=target_sr);
        // 3) Extract model inputs (7s cap ⇒ 112,000 samples)
        inputs ← self.processor( audio,;
        sampling_rate=target_sr,;
        padding=True,;
        truncation=True,;
        max_length=112000,;
        return_tensors='pt' );
        // 4) Tokenize labels with fixed length
        labels ← self.processor.tokenizer( sentence,;
        padding='max_length',;
        max_length=self.max_label_length,;
        truncation=True,;
        return_tensors='pt' )["input_ids"];
        // 5) Mask unwanted tokens:  map to -100 (ignored by CTC loss)
        labels[labels == 54] ← −100                    // empty space;
        labels[labels == 53] ← −100                    // dots;
        // 6) Return tensors (batch dimension squeezed)
        return { "input_values": inputs["input_values"].squeeze(0),;
        "attention_mask": inputs["attention_mask"].squeeze(0),;
        "labels": labels.squeeze(0) };
```